Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Together. Free your energies

The Capgemini Approach to Sizing Non-Custom Software Development Projects/Project Elements

Carl Bideau

14th October 2010

THE Open GROUP
Master
Certified IT Specialist

CONTENTS

- **1. Recap on sizing of custom software development projects**
- 2. How to size non-CSD projects (e.g. Application Integration)?
- 3. The challenge of ‘hybrid’ projects including non-CSD elements
- 4. Key principles
- 5. Questions & Discussion

1. Recap on Sizing of CSD Projects:
Options for CSD Sizing Measures (Why not FP?)

- Industry-standard measure (e.g. Function Points)

  versus

- In-house-developed measure (e.g. Capgemini Use Case Points)

	Advantages	Disadvantages
Function Points Count	▪ An industry standard	▪ Requires expertise in FP counting ▪ Expensive to count (2-3% of budget) ▪ Non-intuitive empirical measure that has no meaning & some odd rules ▪ Not easy to use at bid stage ▪ Terminology & concepts do not map well on to modern tools & methods
Capgemini Use Case Points Count	▪ Available early in project ▪ Relatively cheap & easy to use ▪ Fits in with Use Case method for requirements capture ▪ Has meaning throughout lifecycle	▪ No industry guidelines so cannot be used for productivity 'benchmarking' *(but could translate to FP)* ▪ Measure functional size only *(but non-funcs are a productivity factor!)*

- **Mark I FPs are fairly common within the industry & some industry metrics available**
- **In Capgemini we have found that Capgemini Use Case Points work best for us**

1. Recap on Sizing of CSD Projects:
Capgemini Use Case Points (UCPs)

What are Capgemini use case points?

- In software engineering, use cases are just a structured way of capturing the user requirements
- Capgemini use case points (UCP) are a means by which the functional content of an application's use cases can be quantified / sized in points
- UCPs are an abstract size measure calculated by:
  - Quantifying the number & complexity of Use Case components:
    Flows, Boundary Classes & Business Rules
  - Associating each measure total with their own points scale
  - Applying an average across the individual measured totals to arrive at an overall points total for the Use Case
- A typical small use case equates to approximately 1 UCP!

Problem:
FP & UCP measures only work for sizing Custom Software Development !

You can find a copy of the Capgemini Use Case Sizing Guidelines template on the UKSMA website at: http://www.uksma.co.uk/capgeminiUCP.asp

CONTENTS

- 1. Recap on sizing of custom software development projects
- **2. How to size non-CSD projects (e.g. Application Integration)?**
- 3. The challenge of ‘hybrid’ projects including non-CSD elements
- 4. Key principles
- 5. Questions & Discussion

2. How to Size Non-CSD Projects (e.g. Application Integration)?
Use of Application Integration as an Example

- The sizing of Application Integration elements in a project requires a different sizing approach to that used for custom software development.
- **In this presentation Application Integration is used as an example of how Capgemini has sized project types other than custom software development.**
- Definition:
  - *In this context*: Application Integration refers to the technical mechanisms for transmitting data/messages from one application to another but excludes the business logic for generating and processing the message in each of the end point systems.

2. How to Size Non-CSD Projects (e.g. Application Integration)?
Capgemini Application Integration Sizing Method

- In Capgemini we do not model the technical detail
  - i.e. Ignore the (many & changing!) particular toolsets & packages to be used
- Instead we capture the AI scope
  - i.e. Consider the Integration patterns to be delivered
- We do this by quantifying & defining the complexity of the following building blocks:
  - **Service / Adapter** – the technology to connect the Endpoint to the Integration environment to send / receive data. Not the business logic of the application/service itself
  - **Transport** – the physical means of sending and receiving data from one end point to another e.g. TCP/IP, FTP over TCP/IP, Message queues & channels, VPNs, leased lines.
  - **Dataflow / Mediation** – mediation product intervention.

2. How to Size Non-CSD Projects (e.g. Application Integration)?

Integration Stack Elements/Building Blocks

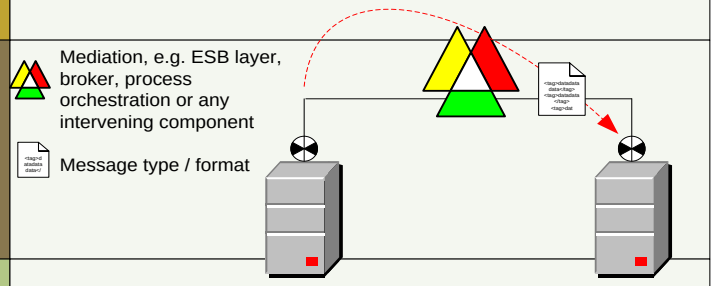
This shows the diagrammatic conventions!

Business Process Management
Short-lived, fully automated process orchestration through to longer lived, partial human-interaction work flow

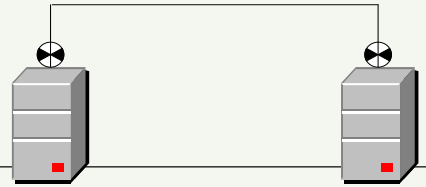
For large-scale BPM, estimate this as a separate project - do not use the AI sizing approach. For smaller-scale process orchestration, specify additional Dataflows to represent processes.

For ad hoc small scale process orchestration, include additional Dataflows.

Dataflow / Mediation
This represents all activity relating to the mediation of data, i.e. all processing applied to data from the point it leaves its source to the point it reaches its target. This includes Data Transformation, small scale Process Orchestration, Routing rules, Service Registry access, cross reference lookups as well as message format definitions – XML / XSD structures - SOAP, canonical message formats etc

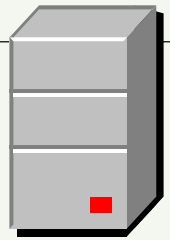


Transport
This is the physical means by which data gets to or from the Service / Adapter to the rest of the Integration environment. Examples include Message Queue channels, VPN tunnels,



Service / Adapter
The term 'Service/Adapter' means the technology responsible for getting data into or out of an Endpoint System. This may be a Web Service definition (WSDL), a file based mechanism, database connectivity, a put or get operation on a queue etc. It is not the business logic of the application or service itself: it is the interfacing code only.

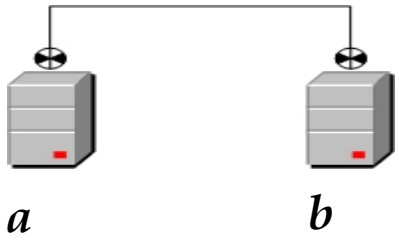
Endpoint Systems / Applications
the Applications to be Integrated, including the CSD development itself, if applicable



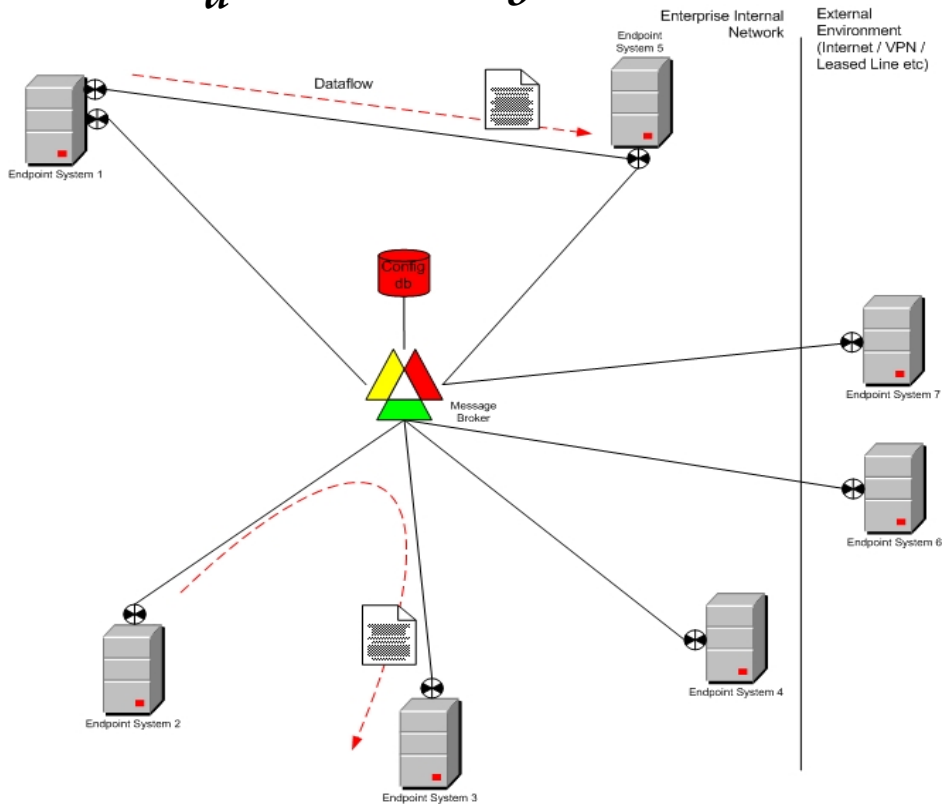
A Dataflow = a Message sent from one Endpoint system to another in one direction

2. How to Size Non-CSD Projects (e.g. Application Integration)?

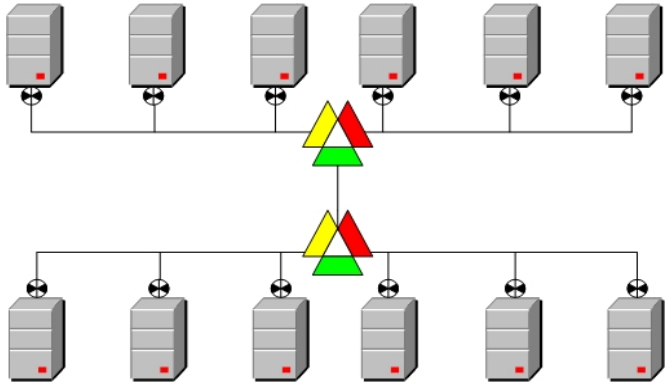
Example Application Integration Patterns



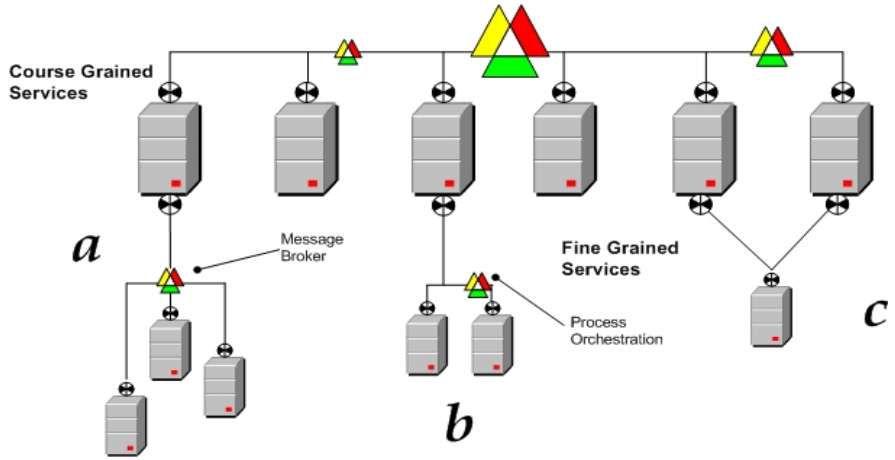
Request / Reply



EAI style Hub & Spoke Environment



Federated ESB / Brokers



Coarse Grained SOA services, comprised of Fine Grained internal services

2. How to Size Non-CSD Projects (e.g. Application Integration)?
Building Block Complexity

- List the endpoint systems together with their associated Adaptors & Transport requirements.
- Separately list the required dataflows.
- Use complexity factors to adjust the size of each of the identified building blocks:
  - **Service / Adapter** – Increase the complexity to:
    - Represent additional tooling and therefore effort in what is done to get an application sending or receiving data
    - No adapter work = 100% reuse, but still select the complexity for documentation
  - **Transport** – Increase complexity to acknowledge (e.g.):
    - Message queuing effort
    - Network layer work – e.g. firewalls, ISP config
  - **Dataflow / Mediation** – Increase complexity to acknowledge that the following products are being used (e.g.):
    - Message Brokers
    - ESB central components
    - Service Registries
    - Process Orchestration (Simple to medium)
    - Effort includes definition of message formats

2. How to Size Non-CSD Projects (e.g. Application Integration)?
Summary of Capgemini AI Points Calculation

- **AI Points Calculation**

Activity – Customise adapters:

Total Adaptor Points = Σ of points for each Adaptor

Activity - Develop / configure transport mechanism:

Total Transport Points = Σ of points for each Transport

Activity - Maintain Interface catalogue + Produce common data formats (Canonical):

Total Dataflow Points = Σ of points for each Dataflow

Total AI Points = Adaptor Points + Transport Points + Dataflow points

	Design/Implementation/Testing Size in AI points		
	Adaptors	Transport	Dataflows
Simple	3.5	0.7	2.5
Medium	7.0	1.4	5.0
Complex	14.0	2.7	10.0
Very Complex	28.0	5.4	20.0

e.g. a medium complexity Dataflow is 5 Capgemini AI points

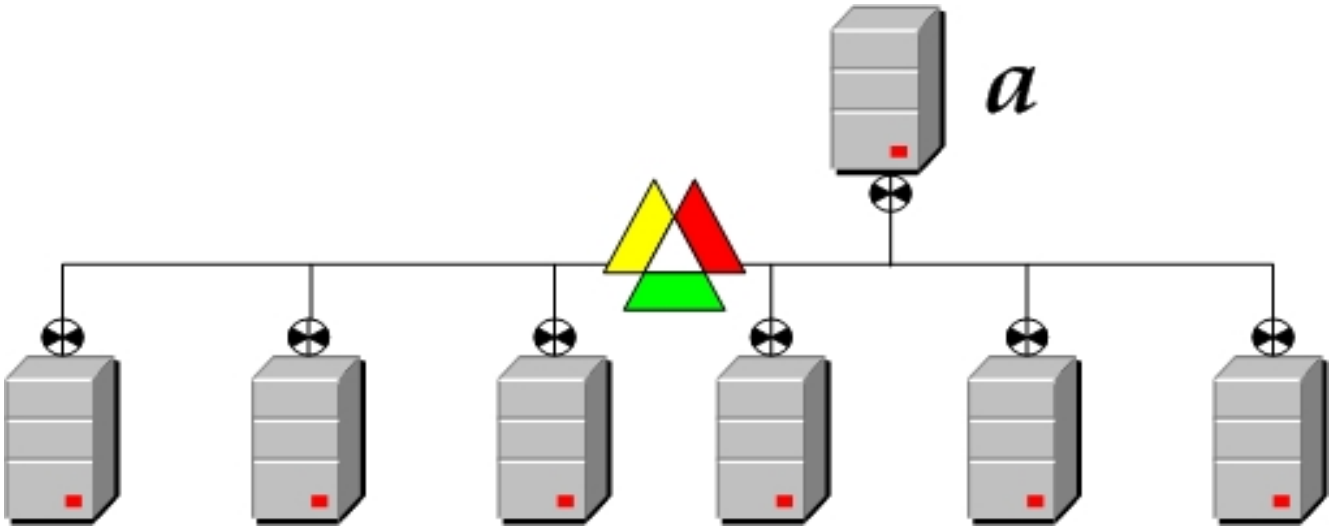
- Add 20% to points for Adaptor if Adaptor is external to client LAN
- Add 20% to points for Transport or Dataflow if Transport or Dataflow traverses the Internet or a WAN.

- **Function Points Approximation**

To convert to Function Point Equivalents multiply AI Points by gearing factor of 3.87

2. How to Size Non-CSD Projects (e.g. Application Integration)?
Example of Publish / Subscribe AI Pattern Sizing

- Endpoint system *a* publishes data to a publish/subscribe engine, which six other endpoint systems subscribe to:



- For the example shown, configure:
  - 7 Endpoint systems, each with 1 Service / Adapter
  - 1 Dataflow in total, which is Endpoint *a* publishing the data

2. How to Size Non-CSD Projects (e.g. Application Integration)?
Example of Publish / Subscribe AI Pattern Sizing

Endpoint System Name	Service / Adapter Name	Internal or External	Development Location	Ease of Integration	Service / Adapter Complexity	Service / Adapter % Re-use	Transport Complexity	Transport % Re-use
Publishing System a	Adapter a	Internal	Offshore	Medium	Complex		Medium	
Subscribing System #1	Adapter #1	Internal	Offshore	Medium	Complex		Medium	
Subscribing System #2	Adapter #2	Internal	Offshore	Medium	Complex	50%	Medium	50%
Subscribing System #3	Adapter #3	Internal	Offshore	Medium	Complex	50%	Medium	50%
Subscribing System #4	Adapter #4	Internal	Offshore	Medium	Complex	50%	Medium	50%
Subscribing System #5	Adapter #5	Internal	Offshore	Medium	Complex	50%	Medium	50%
Subscribing System #6	Adapter #6	Internal	Offshore	Medium	Complex	50%	Medium	50%

- The complexities of the Service / Adapter and Transport of the 6 subscribers account for their subscription calls

Source System name (Optional)	Dataflow/Mediation Name	Target System name (Optional)	Number of Occurrences	Internal or External	Development Location	Dataflow/Mediation Complexity	Dataflow/Mediation % Re-use	Comments
	Publish / Suscribe (system a to systems 1 to 6)		1	Internal	Offshore	Medium		

2. How to Size Non-CSD Projects (e.g. Application Integration)?
Requirements Size for Adapters & Transport

- The size of Requirements work may not be the same as the size of Design/Implementation/Test since non-technical factors can make it simple/difficult to ascertain the requirements
- For Adapters & Transport:
  - **Requirements size = (Requirements Complexity) x (Adapter or Transport size)**

- **Requirements complexity**: Extent to which it is straightforward/complex to work with endpoint system owner to get info. & authorization needed to establish a connection with their system.

  Simple
  - E.g. System owner is internal to client organization & has experience with interfacing their system.

  Medium
  - E.g. Private sector client who is motivated to co-operate.

  Complex
  - E.g. Workshops will be required to agree the interface definitions - no other regular contact is feasible.

  Very Complex
  - E.g. External system itself is being changed to support another project's requirements.

	Requirements Complexity
Simple	3.5
Medium	7.0
Complex	14.0
Very Complex	28.0

See slide notes for further examples

2. How to Size Non-CSD Projects (e.g. Application Integration)?
Integration Scenarios (Patterns)

- Using the building blocks approach it is possible to model most Integration scenarios (patterns):
  - Request / Reply
  - Publish / Subscribe
  - Batch File Transfer
  - Web Services
  - Coarse Grained SOA Services
  - EAI style Hub and Spoke Environment
  - Services Registry / Repository / Adaptive Composite Services
  - Federated ESB / Brokers
  - Canonical Message Formats
- Includes legacy approaches through to contemporary & strategic approaches!

The more the estimator knows about the proposed implementation, the more accurate the estimate

CONTENTS

3. The Challenge of 'Hybrid' Projects
Sizing Units: 'Horses for Courses'

- There is no single sizing unit that works well for all types of project:
  - Use Case Points or Function Points are fine for measuring custom software development but no one would dream of using them for a large scale SAP implementation!
- We have seen that Capgemini Use Case Points work well for CSD projects but they don't measure the right characteristics to be useful for AI projects
  - Solution: We use Capgemini AI points instead!

There is currently no single sizing method that will work for all project types - Each type of project requires its own specific sizing measure!

3. The Challenge of 'Hybrid' Projects

Sizing of Hybrid Projects

- Hybrid projects contain >1 project type such as CSD+AI (*or SAP+CSD+BI etc.*)
  - CSD element sized in Capgemini Use Case Points (UCPs) or sometimes in FPs
  - Application Integration part is sized in Capgemini AI points

BUT:

- We need a single measure of size for the total project

SOLUTION:

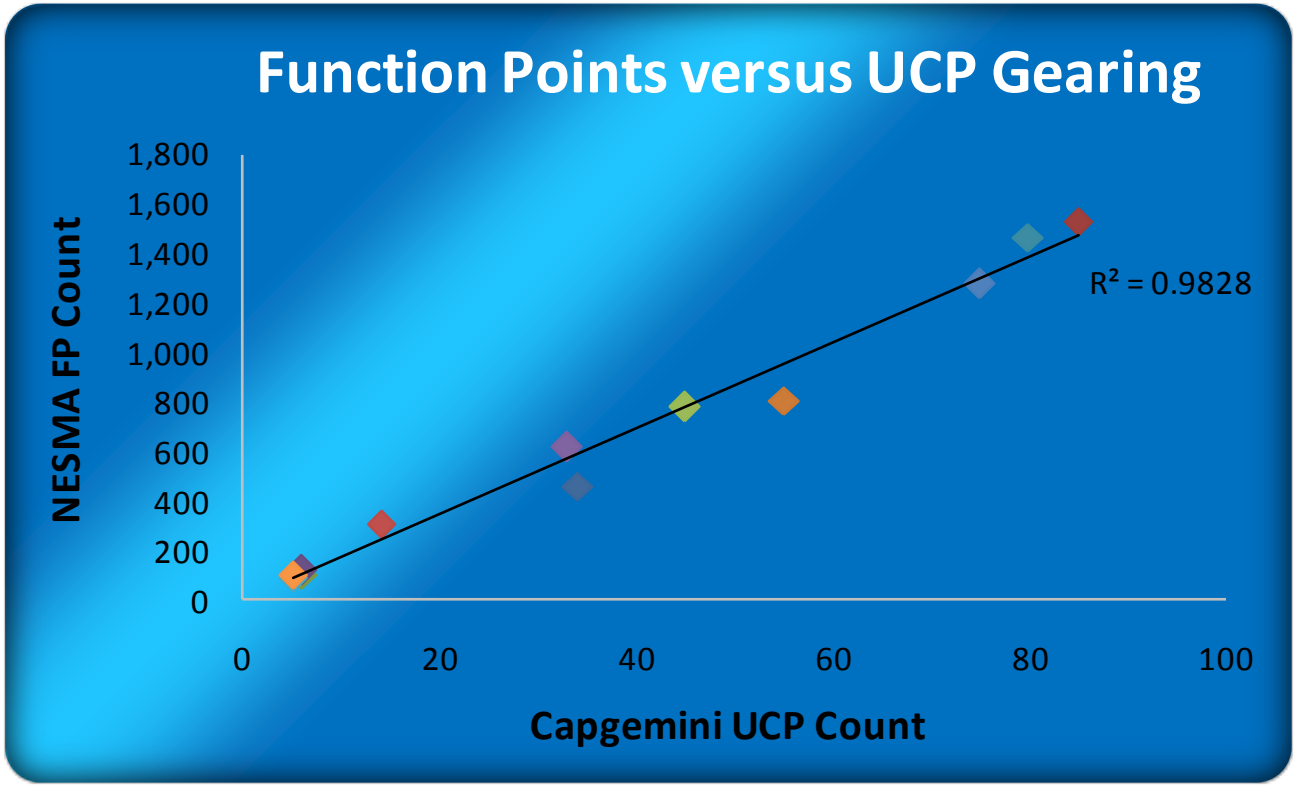
- Size each project type using units applicable to that type
- Use gearing factors to translate each into a common sizing unit (FP equivalents):
  - **Project Size = CSD Functional Size + AI Functional Size**

Note: Non-functional requirements size is addressed via productivity factors

3. The Challenge of 'Hybrid' Projects

Work Still to Do 1

- Continue to analyse current relationship between UCPs & FPs and update/derive accurate UCP to FP gearing factors:



- **The chart above plots the relationship between NESMA Function Points and Capgemini Use Case Points using 2008 Capgemini Dutch data.**
- **The chart shows that here is a good correlation between the Capgemini use case sizing and Function Points giving confidence that the UCP method works.**

3. The Challenge of 'Hybrid' Projects
Work Still to Do 2

- Define reliable SLOC to FP gearing factors
  - So that we can size existing applications easily using a SLOC count
  - Some challenges with reliability of published SLOC to FP gearings!
- Finalise sizing measures for the following project types:
  (*including gearing factors to convert to FP equivalents*)

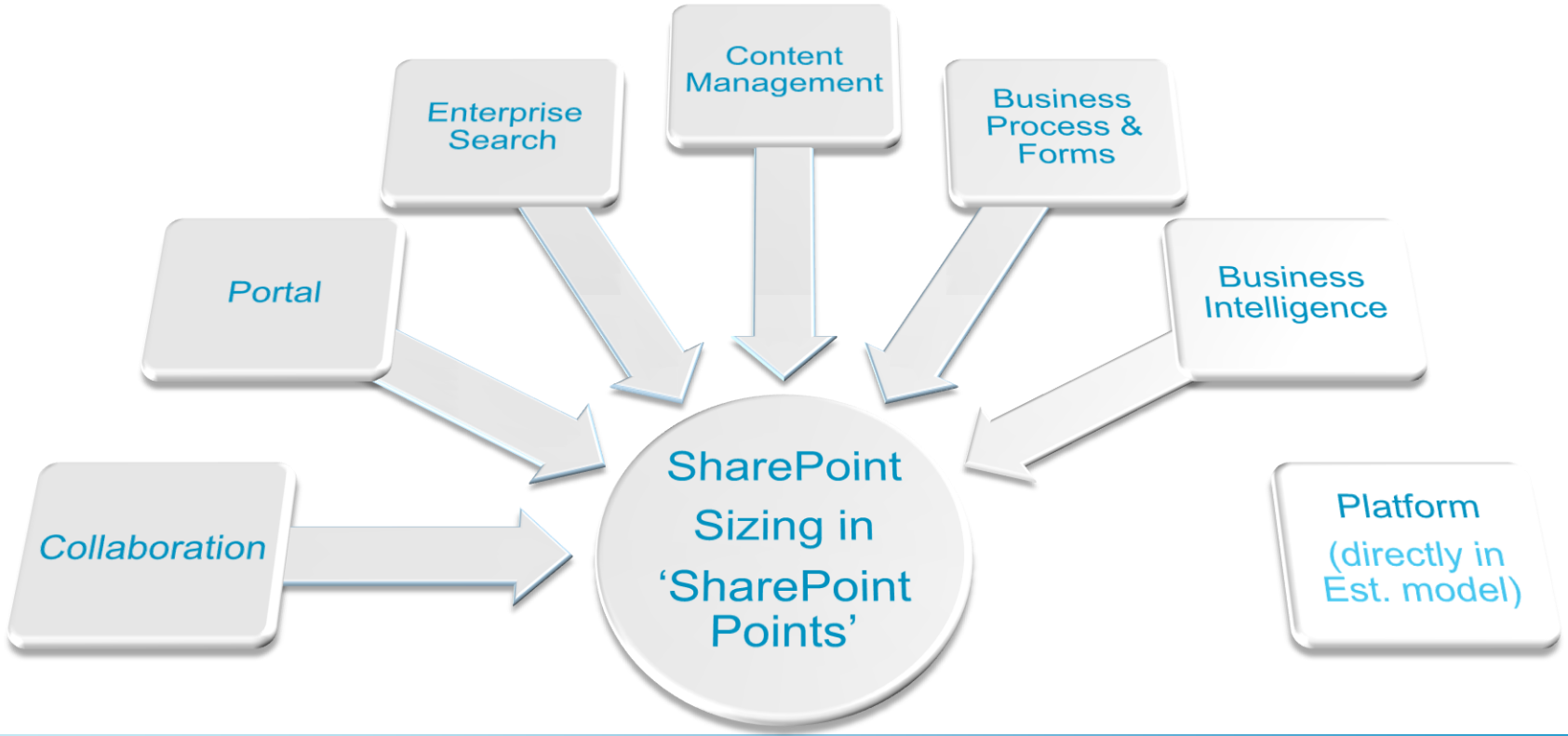
• RUP CSD	– **Capgemini UCP**	- *complete*
• Application Integration	– **Capgemini AI points**	- *complete*
• SAP Implementation	– **Capgemini SAP points**	- *early draft*
• Oracle Implementation	– **Capgemini Oracle points**	- *early draft*
• Business Intelligence	– **Capgemini BI points**	- *nearly complete*
• Data Migration	– **Capgemini DM points**	- *complete*
• Information Management	– **Capgemini IM points**	- *SharePoint element complete*

3. The Challenge of 'Hybrid' Projects

Example of Another Project Type: SharePoint

Taking the Capgemini CSD Use Case Sizing Guidelines as a 'model' an Excel spreadsheet was developed for sizing SharePoint

This takes data from 7 core input areas & generates an overall sizing in 'SharePoint Points' (SP)



CONTENTS

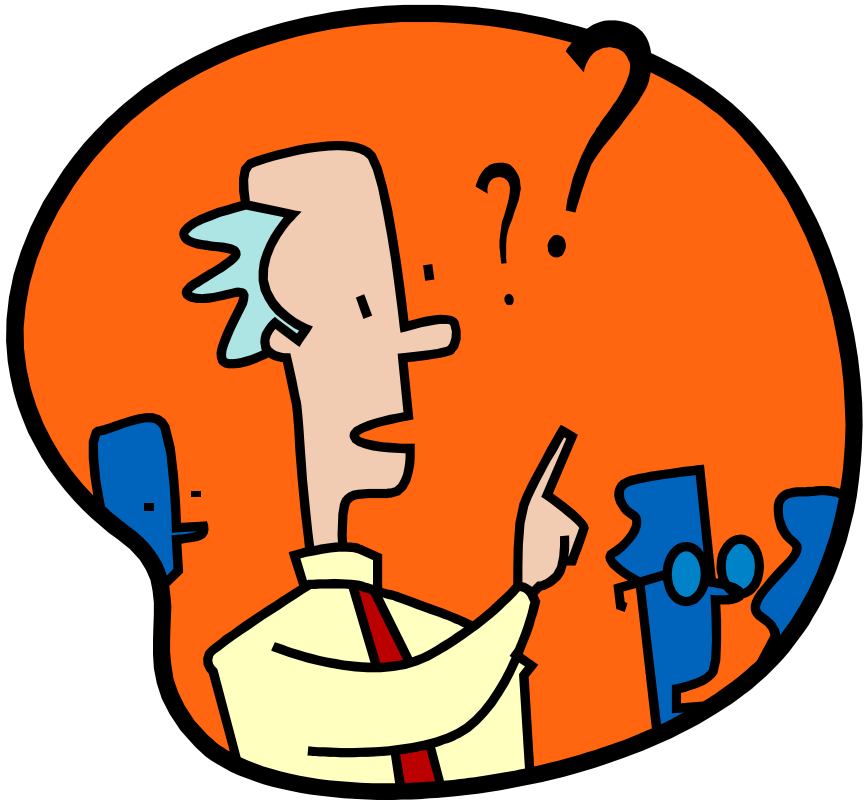
4. Key principles
Sizing Units for Productivity Measurement

- Using gearing factors to translate between sizing units compromises accuracy – it is only ever an approximation!
- To **estimate project effort** it is safest to use the most appropriate sizing unit for that project type
  - Sometimes we cannot avoid converting into common sizing units to calculate a total size for the project
- For **retrospective productivity measurement** the same issue applies: it will be most accurate to calculate productivity in terms sizing units that are most relevant for that project type
  - In any case why would you want to compare the productivity of a CSD project with an AI one? This is like comparing apples with oranges!
- Hybrid projects (e.g. CSD + AI) are a problem: the only options are:
  - Convert the sizes to common sizing units (e.g. FP equivalents)
  - Group projects: e.g. into ‘Pure CSD’, ‘Pure AI’, ‘Hybrid’ for comparison purposes

CONTENTS

- 1. Recap on sizing of custom software development projects
- 2. How to size non-CSD projects (e.g. Application Integration)?
- 3. The challenge of ‘hybrid’ projects including non-CSD elements
- 4. Key principles
- **5. Questions & Discussion**

5. Questions & Discussion



Capgemini

CONSULTING.TECHNOLOGY.OUTSOURCING

Together. Free your energies

www.capgemini.com